Amendments to the specification are indicated in the attached "Marked Up Version of Amendments" (pages i - iv).

In the Claims

Please amend Claims 4, 6, and 12. Amendments to the claims are indicated in the attached "Marked Up Version of Amendments" (page iv-v).

A9

4. (Amended) A file subsystem as in claim 1 wherein the file subsystem is incorporated in one of a Web page home server or Web page cache server.

A10

6. (Amended) A file subsystem as in claim 5 wherein the directory entry specifies at least a partition, starting block number, and size for the data object.

A12

12. (Amended) A storage subsystem as in claim 1 wherein a new data object is written to overwrite a previously stored data object in a segment based on a time stamp indicating when the previous data object was stored.

Please add new Claims 13-72.

A12 Cont

13. (NEW) A method of maintaining data objects in a memory device of a web cache server, the method comprising:

partitioning the memory device to include segments of memory space, at least one of the segments being used to store multiple data objects;

determining that a new data object is to be stored in the memory device;

freeing up memory space in a segment of the memory device allocated for storage of multiple data objects, the freed up memory space previously being used to store multiple data objects; and

storing the new data object in at least a portion of the freed up memory space.

14. (NEW) A method as in claim 13 further comprising:

utilizing a linear search based on a hashed file specifier to locate and retrieve an object stored in the memory device.

15. (NEW) A method as in claim 13, wherein the memory device is a disk storage device including rings partitioned to include multiple segments.

16. (NEW) A method as in claim 15, wherein multiple segments of the disk storage device are fixed and similar in size.

17. (NEW) A method as in claim 13, wherein at least two segments of the memory device are of equal size.

A12 Cm.1

18. (NEW) A method as in claim 13 further comprising:

storing multiple data objects in a segment, at least two of which are contiguous with each other.

19. (NEW) A method as in claim 13, wherein at least two data objects stored in a given segment vary in size with respect to each other.

20. (NEW) A method as in claim 13 further comprising:

storing a variable number of data objects in segments of the memory device.

21. (NEW) A method as in claim 13, wherein each of the multiple data objects stored in the memory device is not partially stored in any segment.

22. (NEW) A method as in claim 13 further comprising:

storing new data objects to the memory device in a round-robin fashion.

23. (NEW) A method as in claim 13 further comprising:

maintaining a directory table associated with the data objects in the memory device in a volatile memory device; and

hashing a file specifier associated with a requested data object to locate the requested data object in the memory device.

24. (NEW) A method as in claim 13, wherein an oldest stored segment of data objects in the memory device is overwritten with newly stored data objects.

25. (NEW) A method as in claim 24 further comprising:

providing a time stamp associated with data objects stored in the memory device to determine which of multiple segments will be freed to provide space for storing new data objects.

A12 cm.T

26. (NEW) A method as in claim 13, wherein a segment includes at least part of a ring of the memory device.

27. (NEW) A method as in claim 13, wherein the data objects, once stored in a corresponding segment, are not modified until the previously stored data objects become freed up memory space.

28. (NEW) A method as in claim 13, wherein a data object includes an object header and object trailer comprising a string of bits that are used to verify the data object with information stored in a directory table, the directory table being searched to find the data object in the memory device.

29. (NEW) A method as in claim 13, wherein a segment of expired data objects in the memory device is overwritten with newly stored data objects.

30. (NEW) A method of maintaining data objects in a web cache server, the method comprising:

partitioning a memory device associated with the web cache server into multiple segments;

storing data objects transmitted over a network in segments of the memory device, filling the memory device to capacity with segments of data objects that are not modified after initial storage;

detecting that a new data object is to be stored in the memory device; and

storing the new data object in a particular segment of the memory device overwriting previously stored data objects depending on a time when the data objects filling the memory device were previously stored.

A13 Cont

31. (NEW) A method as in claim 30, wherein the new data object is written over memory space of expired data objects.

32. (NEW) A method as in claim 30, wherein the new data object is written in memory space that previously stored oldest data objects in the memory device.

33. (NEW) A method as in claim 30, wherein the step of storing the new data object further includes:

identifying a next available contiguous free space in the particular segment; and

determining whether the new data object, if stored in the particular segment, can be stored without overlapping a segment partition.

34. (NEW) A method as in claim 33 further comprising:

storing the new data object in a following contiguous segment of the memory device if the new object cannot be stored in its entirety in the particular segment.

35. (NEW) A method as in claim 30 further comprising:

partitioning a segment of the memory device to include free space to store multiple data objects of variable lengths.

36. (NEW) A method as in claim 35, wherein the step of storing the new data object further includes:

storing the new data object to be contiguous with a previously stored data object in the particular segment.

37. (NEW) A method as in claim 30, wherein the memory device is a ring that is partitioned into the multiple segments.

38. (NEW) A method as in claim 30, wherein the step of storing the new data object further includes:

identifying a next available contiguous free space in the particular segment; and

storing the new data object to be contiguous with a recently stored data object in the particular segment.

A13 Cont

39. (NEW) A method of maintaining data objects in a memory device of a web cache server, the method comprising:

partitioning the memory device into segments, at least one of the segments being used to store multiple data objects of varying size;

determining that a new data object is to be stored in the memory device;

identifying free memory space in an active segment presently being filled with data objects; and

writing the new data object in the free memory space, the new data object being stored contiguous with a last data object written to the active segment.

40. (NEW) A method as in claim 39 further comprising:

detecting that a second new data object is to be stored in the in the memory device;

identifying that the second new data object is larger than free memory space available in the active segment; and

selecting a new segment to store the second new data object.

41. (NEW) A method as in claim 40, wherein the new segment for storing the second new data object is contiguous with the last active segment for storing data objects.

42. (NEW) A method as in claim 39, wherein new data objects stored in the active segment vary in size.

43. (NEW) A method as in claim 39 further comprising:

scanning information associated with the data objects in the memory device to determine which of multiple previously written segments will be selected as the active segment to store new data objects.

44. (NEW) A method as in claim 43, wherein the information includes time stamps associated with data objects stored in the memory device.

A12 cont

45. (NEW) A data storage and retrieval method comprising the steps of:

providing a first memory device in a data cache server for storing and retrieving web data objects;

partitioning the first memory device to include segments for storing data objects of varying size, the data objects being stored sequentially and contiguously in a segment but not overlapping from one segment to the next;

maintaining a directory table including a plurality of blocks in a second memory device, the blocks including pointers identifying locations of objects in the first memory device;

intercepting a request for a data object, the request being a network message intended for another node on a network;

determining whether the requested object is in the first memory device and, if so, retrieving the requested object from the first memory device.

46. (NEW) A method as in claim 45, wherein each data object stored in the memory device is stored in a corresponding contiguous memory space.

47. (NEW) A method as in claim 45, wherein a data object includes an object header and object trailer comprising a string of bits that are used to verify the data object with information stored in a directory table.

48. (NEW) A method as in claim 47, wherein the object trailer includes a hash value.

49. (NEW) A method as in claim 47, wherein the object header includes information indicating an object size of a corresponding data object.

50. (NEW) A method as in claim 45 further comprising:

hashing a file specifier associated with the requested object; and

locating a pointer associated with the requested object in the second memory device based on the hashed file specifier.

A12 Cont

51. (NEW) A method as in claim 50, wherein the hashed file specifier includes a first set of bits identifying a block in a directory table that includes a pointer indicating a location of the requested object.

52. (NEW) A method as in claim 51, wherein the file specifier is a URL (Uniform Resource Locator).

53. (NEW) A method as in claim 45, wherein the segments are fixed in size.

54. (NEW) A method as in claim 53, wherein the segments are of equal size.

55. (NEW) A method of maintaining retrievable objects, the method comprising:

providing a first memory device to store data objects;

maintaining a directory table in a second memory device to track the objects stored in the first memory device, the directory table including pointers identifying locations of the objects in the first memory device; and

storing meta data information along with corresponding data objects in the first memory device;

storing meta data information associated with the data objects in entries of the directory table; and

ensuring data object integrity of a retrieved data object using the meta data information stored in both the first memory device and the directory table.

56. (NEW) A method as in claim 55, wherein the first memory device is a mass storage memory disk.

A12 cont

57. (NEW) A method as in claim 55 further comprising:

intercepting a request for a data object at an intermediate node of a network, the request being a network message intended for another node on a network; and

determining whether the requested object is in the first memory device and, if so, retrieving the requested object from the first memory device.

58. (NEW) A method as in claim 55 further comprising:

partitioning the first memory device to include segments for storing data objects of varying size, the data objects being stored sequentially and contiguously in a segment but not overlapping from one segment to the next.

59. (NEW) A method as in claim 55, wherein the data objects are retrieved from the first memory device without either write ordering or running a pre-startup recovery program.

60. (NEW) A method as in claim 55 further comprising:

detecting a request for a data object; and

matching information stored in the first memory device associated with the requested data object to corresponding information stored in the directory table to ensure object integrity.

61. (NEW) A method as in claim 55, wherein header and trailer information associated with the object stored in the first memory device is used to validate a retrieved object.

62. (NEW) A method as in claim 55 further comprising:

utilizing header or trailer information of an object stored in the first memory device with information entered in the directory table to validate a stored data object.

63. (NEW) A method as in claim 55, wherein the meta data information stored in the first memory device includes URL (Uniform Resource Locator) information.

64. (NEW) A method as in claim 55, wherein the meta data information stored in the first memory device includes hash information.

A13 Cn.4

65. (NEW) A method as in claim 55, wherein the meta data information stored along with an object in the first memory device includes object size information.

66. (NEW) A method as in claim 65 further comprising:

eliminating a need to run a pre-startup recovery program prior to retrieving objects stored in the first memory device.

67. (NEW) A method as in claim 55, wherein a need for index nodes is eliminated.

68. (NEW) A method as in claim 55 further comprising:

detecting that an object is invalid based upon non-matching data stored in the first and second memory device.

69. (NEW) A method as in claim 68, wherein an object is determined to be invalid based upon non-matching hash information.

70. (NEW) A method as in claim further comprising: